



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/761,547	01/20/2004	Arthur E. Alacar	2003-1101	2850
7590	05/02/2008		EXAMINER	
Kyocera Technology Development Intellectual Property Administration Suite 400 1855 Gateway Blvd. Concord, CA 94520			DHINGRA, PAWANDEEP	
			ART UNIT	PAPER NUMBER
			2625	
			MAIL DATE	DELIVERY MODE
			05/02/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/761,547	ALACAR, ARTHUR E.
	Examiner	Art Unit
	PAWANDEEP S. DHINGRA	2625

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 09 January 2008.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-22 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-22 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) Notice of Informal Patent Application

6) Other: _____.

DETAILED ACTION

- This action is responsive to the following communication: Amendment after non-final rejection filed on 01/09/2008.
- Claims 1-22 are pending in the present application.

Response to arguments

Applicant's amendments, filed 01/09/2008 have been entered. However, applicant's arguments, filed 01/09/2008 have been fully considered but they are not persuasive.

Applicant argues that newly amended limitations added to the independent claims are not taught by either Kemp or Nguyen.

In reply, the examiner asserts that Nguyen clearly teaches the newly added limitations of independent claims (see discussion of claim 1 below).

Claim Rejections - 35 USC § 101

Previous 101 rejections to claims are withdrawn in view of applicant's amendments to the claims.

Examiner Notes

Examiner cites particular paragraphs, columns and line numbers in the references as applied to the claims below for the convenience of the applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested that, in preparing responses, the applicant fully consider the references in entirety as potentially teaching all or part of the claimed

invention, as well as the context of the passage as taught by the prior art or disclosed by the examiner.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-12, and 14-22 are rejected under 35 U.S.C. 103 as being unpatentable over Kemp et al., US 2003/0200427 in view of Nguyen et al., US 6,825,941.

Re claim 1, Kemp et al. discloses a method for augmenting a printer driver (see paragraphs 45-47), comprising: providing a GUI (user interface, see figures 2-3, 7-8; paragraph 13) for selecting at least one plug-in module (see S904, figure 9; paragraph 76) (also see figure 2-10); and dynamically adding (installing) the at least one plug-in module to the printer driver (see S905-S910; paragraphs 75-78).

Kemp fails to disclose providing a heap area for private devmode structures; wherein adding of each of the at least one plug-in module results in allocating and initializing by a printer driver of a private devmode structure in the heap area only when necessary to accomplish loading for UI display and printing, and wherein later removing of each of the at least one plug-in module results in deallocation of the corresponding

private devmode structure in the heap area only when necessary to accomplish loading of a printer driver .

However, Nguyen et al. discloses providing a heap area for private devmode structures (see column 14, line 65-column 21, line 60); wherein adding of each of the at least one plug-in module (column 5, lines 7-30; column 8, lines 4-25; column 9, line 10 - column 10, line 60) results in allocating and initializing by a printer driver of a private devmode structure in the heap area only when necessary to accomplish loading for UI display and printing (see column 14, line 65-column 22, line 26; column 24, line 44-column 25, line 47; Column 30, lines 21-59) and wherein later removing of each of the at least one plug-in module results in deallocation of the corresponding private devmode structure in the heap area only when necessary to accomplish loading of a printer driver (see column 14, line 65-column 22, line 26; column 24, line 44-column 25, line 47; Column 30, lines 21-59).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention to include the extensible driver architecture of Nguyen et al. into the extensible device driver of Kemp et al. for the benefit of allowing “*OEM's to plug in special code for customizing the UI, bitmap handling, font and text processing, and general printer control ...utilizing “a flexible modular architecture which allows enhancements to the driver to be implemented to provide better support for more varieties of output devices, and to improve the output quality, ease of use and performance without the necessity for redesign“* as taught by Nguyen at column 3, lines 20-47.

Re claim 2, Kemp et al. further discloses the adding (installing or entering) of the at least one plug-in module comprises copying at least one plug-in DLL file (dynamic link library file) to a printer system folder (system memory) (see paragraph 75-78 and paragraph 19 & 51-52, note that "*the device driver module and the driver plug-in module are each preferably comprised of a dynamic link library file*", hence it is apparent that for adding new plug-in module to the existing print driver of printer, the plug-in files in the form DLL get copied into the system folder of the printer).

Re claim 3, Kemp et al. further discloses the adding of the at least one plug-in module comprises checking compatibility (checking if plug-in module is supported) of at least one plug-in DLL file with the printer driver (see figures 6A-6B; paragraphs 70-72, note that the found plug-in is checked to see if it is supported (compatible) with the interface of existing printer driver, and if it is than it gets sent (S613-S614) and eventually appears as available in S903 for further processing as shown in figure 9. Also, see paragraphs 18-19 and note that "*the device driver module and the driver plug-in module are each preferably comprised of a dynamic link library file*", hence it is apparent that the DLL files of the plug-in are checked for compatibility (supportiveness) with the interface of printer driver).

Re claim 4. Kemp et al. further discloses the adding of the at least one plug-in module comprises the at least one plug-in module installing itself (see paragraph 76, note that plug-in can be selected automatically (without user interaction), and if at S906 it is also determined by the process flow that no pre-plug-in exists (without user

interaction) then in this case, the process flow will jump to S910, and plug-in module will be installed itself (automatically, without any user interaction).

Re claim 5, Kemp et al. further discloses the adding of the at least one plug-in module comprises adding at least one registry entry (see paragraphs 75-78).

Re claim 6, Kemp fails to further disclose that the adding of the at least one plug-in module comprises heap-allocating and initializing at least one private devmode structure.

However, Nguyen et al. discloses that the adding of the at least one plug-in module to the printer driver (see column 5, lines 7-30; column 8, lines 4-25; column 9, line 10 - column 10, line 60) comprises heap-allocating (allocating memory heap) (see column 20, line 20 - column 21, line 60) and initializing at least one private devmode structure (see column 14, line 65-column 21, line 60).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention to include the extensible driver architecture of Nguyen et al. into the extensible device driver of Kemp et al. for the benefit of allowing “*OEM's to plug in special code for customizing the UI, bitmap handling, font and text processing, and general printer control ...utilizing “a flexible modular architecture which allows enhancements to the driver to be implemented to provide better support for more varieties of output devices, and to improve the output quality, ease of use and performance without the necessity for redesign*“ as taught by Nguyen at column 3, lines 20-47.

Re claim 7, Kemp fails to further disclose the heap is a private devmode area following a public devmode area.

However, Nguyen et al. discloses the heap is a private devmode area following a public devmode area (see column 14, line 65-column 21, line 60).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention to include the extensible driver architecture of Nguyen et al. into the extensible device driver of Kemp et al. for the benefit of allowing “*OEM's to plug in special code for customizing the UI, bitmap handling, font and text processing, and general printer control ...utilizing “a flexible modular architecture which allows enhancements to the driver to be implemented to provide better support for more varieties of output devices, and to improve the output quality, ease of use and performance without the necessity for redesign*“ as taught by Nguyen at column 3, lines 20-47.

Re claim 8, Kemp fails to further disclose the heap is fixed size.

However, Nguyen et al. discloses the heap (memory) is fixed size (see column 38, lines 40-62).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention to include the extensible driver architecture of Nguyen et al. into the extensible device driver of Kemp et al. for the benefit of allowing “*OEM's to plug in special code for customizing the UI, bitmap handling, font and text processing, and general printer control ...utilizing “a flexible modular architecture which allows enhancements to the driver to be implemented to provide better support for more varieties of output devices, and to improve the output quality, ease of use and performance without the necessity for redesign*“ as taught by Nguyen at column 3, lines 20-47.

the output quality, ease of use and performance without the necessity for redesign“ as taught by Nguyen at column 3, lines 20-47.

Re claim 9, Kemp fails to further disclose each of the at least one private devmode structure corresponds to each of the at least one plug-in module added, each of which implements an optional feature selected from the group consisting of feature sets, Page Description Languages (PDLs), and Renders.

However, Nguyen et al. discloses each of the at least one private devmode structure corresponds to each of the at least one plug-in module added (see column 14, line 65-column 21, line 60), each of which implements an optional feature selected from the group consisting of feature sets, Page Description Languages (PDLs) (i.e. PCL), and Renders (see column 3, lines 20-37; column 8, lines 4-25; note that the architecture of Nguyen is very extensible and makes it implement new features including supporting PCL commands).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention to include the extensible driver architecture of Nguyen et al. into the extensible device driver of Kemp et al. for the benefit of allowing “*OEM's to plug in special code for customizing the UI, bitmap handling, font and text processing, and general printer control ...utilizing “a flexible modular architecture which allows enhancements to the driver to be implemented to provide better support for more varieties of output devices, and to improve the output quality, ease of use and performance without the necessity for redesign*“ as taught by Nguyen at column 3, lines 20-47.

Re claim 10, Kemp et al. further discloses providing a GUI (user interface, see figures 2-3, 7-8; paragraph 13) by which a user selects at least one plug-in module (see S904, figure 9; paragraph 76, note that selection can be manual by a user) (also see figures 2-10); and removing (deleting) the at least one plug-in module from the printer driver (see paragraphs 77-78, note that *“in step S906, it is determined whether a pre-existing plug-in module has already been registered in registry 41 which is of the same type, or same name, as the selected plug-in module. If so, flow passes to step S907 in which the user of computer 10 is notified of the situation, and it is then determined in step S908 whether the user has instructed to proceed with installation of the selected plug-in module by replacing, or renaming, the pre-existing plug-in module. If the user opts for replacement (or renaming), flow passes to step S909 in which the pre-existing plug-in module is deleted or renamed, as the case may be”*).

Re claim 11, Kemp fails to further disclose the removing of the at least one plug-in module comprises deallocating at least one private devmode structure.

However, Nguyen et al. discloses the removing (replacing) of the at least one plug-in module (plug-in nodes) comprises deallocating (disposing) at least one private devmode structure (heap, note that devmode structures are allocated in the memory heap) (see column 20, lines 20-37; column 16, lines 42-63).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention to include the extensible driver architecture of Nguyen et al. into the extensible device driver of Kemp et al. for the benefit of allowing *“OEM's to plug in special code for customizing the UI, bitmap handling, font and text processing, and general printer control ...utilizing a flexible modular architecture which allows enhancements to the driver to*

be implemented to provide better support for more varieties of output devices, and to improve the output quality, ease of use and performance without the necessity for redesign“ as taught by Nguyen at column 3, lines 20-47.

Re claim 12, Kemp et al. further discloses the at least one plug-in module is stored at a remote storage (server 30's fixed disk) on the network (see paragraphs 42, and 53).

Re claim 14, Kemp et al. further discloses the adding of the at least one plug-in module comprises adding at least one GUI (user interface) tab for the added (detected) at least one plug-in module (see figures 7-8 & 10; paragraph 80).

Re claim 15, claim 15 recites identical features, as claim 1, except claim 15 merely deals with executing the method of claim 1 on a computer. Thus, arguments made for claim 1 are applicable for claim 15.

Re claim 16, claim 16 recites identical features, as claims 2, 3, and 5, except claim 16 merely deals with executing the method of claims 2, 3, and 5 on a computer. Thus, arguments made for claims 2, 3, and 5 are applicable for claim 16.

Re claim 17, claim 17 recites identical features, as claim 6, except claim 17 merely deals with executing the method of claim 6 on a computer. Thus, arguments made for claim 6 are applicable for claim 17.

Re claim 18, claim 18 recites identical features, as claim 12, except claim 18 merely deals with executing the method of claim 12 on a computer. Thus, arguments made for claim 12 are applicable for claim 18.

Re Claim 19, claim 19 recites identical features, as claim 1, except claim 19 is an apparatus claim. Thus, arguments made for claim 1 are applicable for claim 19.

Re Claim 20, claim 20 recites identical features, as claims 2, 3, and 5, except claim 20 is an apparatus claim. Thus, arguments made for claims 2, 3, and 5 are applicable for claim 20.

Re Claim 21, claim 21 recites identical features, as claim 6, except claim 21 is an apparatus claim. Thus, arguments made for claim 6 are applicable for claim 21.

Re Claim 22, claim 22 recites identical features, as claim 12, except claim 22 is an apparatus claim. Thus, arguments made for claim 12 are applicable for claim 22.

3. Claim 13 is rejected under 35 U.S.C. 103 as being unpatentable over Kemp et al., US 2003/0200427 in view of Nguyen et al., US 6,825,941 further in view of Nakao, US 2002/0035941.

Re claim 13, Kemp et al. further discloses the adding of the at least one plug-in module comprises checking at least one registry entry (registry 41) for at least one added (registered) plug-in module (see figure 9; paragraphs 77-78); and copying at least one DLL file corresponding (relating) to the at least one plug-in module (plug-in module is comprised of DLL files (as explained above, earlier), hence DLL files relate to plug-in module) from a server (server 30) to a client (computer 10) (see paragraph 42).

Kemp fails to explicitly disclose copying at least one file corresponding to the added at least one plug-in module from a server to a client

However, Nakao discloses copying at least one file corresponding to the added at least one module from a server to a client (see paragraphs 67-68, note that if there has been change in the registration settings file (i.e. change also could be due to a addition of a module, for instance), then upon user's request, the updates are downloaded from the server to the client corresponding to the change (i.e. added module)).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention to include the extensible driver architecture of Nguyen et al., and the data processing apparatus of Nakao into the extensible device driver of Kemp et al. for the benefit of allowing “*OEM's to plug in special code for customizing the UI, bitmap handling, font and text processing, and general printer control ...utilizing “a flexible modular architecture which allows enhancements to the driver to be implemented to provide better support for more varieties of output devices, and to improve the output quality, ease of use and performance without the necessity for redesign*“ as taught by Nguyen at column 3, lines 20-47, and “*registering print-settings on a printer driver which is a software for controlling a printer*“ as taught by Nakao at paragraph 1.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to PAWANDEEP S. DHINGRA whose telephone number is (571)270-1231. The examiner can normally be reached on M-F, 9:30-7:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Twyler L. Haskins can be reached on 571-272-7406. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/P. D./
Examiner, Art Unit 2625

/Twyler L. Haskins/
Supervisory Patent Examiner, Art Unit 2625